AIAA SciTech Forum
7-11 January 2019, San Diego, California
AIAA Scitech 2019 Forum

10.2514/6.2019-2074

Development of a Peripheral-Central Vision System for Small UAS Tracking

Changkoo Kang*, Haseeb Chaudhry†, Craig A. Woolsey‡ and Kevin Kochersberger§
Virginia Tech, Blacksburg, VA 24061, USA

With the rapid proliferation of small unmanned aircraft systems (UAS), the risk of mid-air collisions is growing, as is the risk associated with the malicious use of these systems. Airborne Detect-and-Avoid (ABDAA) and counter-UAS technologies have similar sensing requirements to detect and track airborne threats, albeit for different purposes: to avoid a collision or to neutralize a threat, respectively. These systems typically include a variety of sensors, such as electro-optical or infrared (EO/IR) cameras, RADAR, or LiDAR, and they fuse the data from these sensors to detect and track a given threat and to predict its trajectory. Camera imagery can be an effective method for detection as well as for pose estimation and threat classification, though one cannot resolve range to a threat from a single camera image without additional information, such as knowledge of the threat geometry. To support ABDAA and counter-UAS applications, we consider a merger of two image-based sensing methods that mimic human vision: (1) a "peripheral vision" camera (i.e., with a fisheye lens) to provide a large field-of-view and (2) a "central vision" camera (i.e., with a perspective lens) to provide high resolution imagery of a specific target. Beyond the complementary ability of the two cameras to support detection and classification, the pair form a heterogeneous stereo vision system that can support range resolution. This paper describes the initial development and testing of a peripheral-central vision system to detect, localize, and classify an airborne threat and finally to predict its path using knowledge of the threat class.

I. Introduction

Cameras are common sensors for Airborne Detect-and-Avoid (ABDAA) and mobile counter-UAS systems because of their low power consumption, light weight, and low cost. A typical application is visual detection, but the narrow field of view (FOV) of a perspective camera is limiting. Drulea *et al*. [1] and Kita *et al*. [2] proposed the use of a stereo "fisheye" vision system in order to relax the FOV limitation, but fisheye cameras provide lower pixel coverage in a given region, making it more difficult to classify a detected threat or to estimate its pose for trajectory prediction at greater distances.

Here, we suggest a peripheral-central vision system that detects, localizes, classifies, tracks and predicts the motion of small UAS for ABDAA or counter-UAS applications. Similar recent efforts include the work of Siewert *et al*. [3], who suggested a ground-based local sensor network that includes an EO/IR sensor, RADAR and ADS-B for small UAS traffic management. Laurenzis *et al*. [4] proposed ground-based and mobile sensing systems to detect and track UAS at low altitude. Chen *et al*. [5] described an object tracking algorithm for a UAS with multiple threat sensors. In these systems, camera imagery is used only to detect threats; we propose to use imagery to infer threat position and attitude, to classify the threat, and to predict the future path of the threat.

To classify a threat aircraft and estimate its pose, in order to better predict its flight path, a higher resolution image is required, as might be obtained from a narrow FOV perspective ("central vision") camera. To provide continuous visual coverage of the environment for threat detection, however, requires a wide FOV ("peripheral vision") camera. Incorporating each type of camera affords an opportunity to use stereo vision for ranging. Accordingly, this paper introduces a heterogeneous "peripheral-central" vision system for a ground- or air-based detect-and-avoid (DAA) system

*Graduate Research Assistant, Department of Aerospace and Ocean Engineering
†Graduate Research Assistant, Department of Mechanical Engineering
‡Professor, Department of Aerospace and Ocean Engineering, AIAA Associate Fellow
§Associate Professor, Department of Mechanical Engineering, AIAA Associate Fellow

that is capable of detecting flying objects within a wide FOV, confirming and classifying these threats, estimating their position (including range), velocity, and pose, and predicting their flight path. Eynard *et al* [6–8] suggested an algorithm to estimate the altitude and motion of an unmanned aircraft using an onboard, heterogeneous stereo vision system that consists of a fisheye lens camera and a perspective camera. Their algorithm first finds the homography matrix relating the the two camera views and then estimates the distance between the horizontal plane (i.e., the tangent plane to the earth's surface) and the first camera. The algorithm determines the altitude of the imaged aircraft, but not the range and does not predict the future path of the threat.

This paper describes initial proof-of-concept analysis and testing, where the peripheral vision camera first detects a threat and then uses the viewing angle to cue the central vision camera, which then slews into position to focus on the threat. The flow chart in Figure 1 provides an overview of the concept; the manuscript describes the system following the order of this flow chart: detect (Section III), stare (Section IV), range (Section V), classify (Section VI) and predict (Section VII). Hardware for a prototype system and the experiment setup are described in Section II. Section VIII presents initial conclusions and summarizes ongoing work toward a complete detection, classification, localization, and prediction scheme.

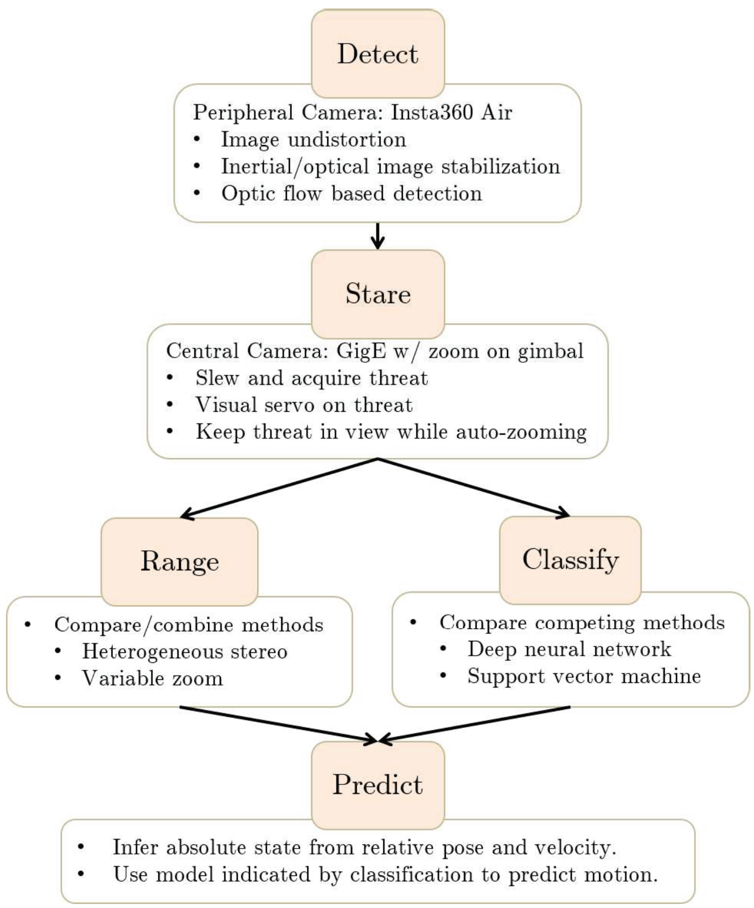


Fig. 1 The peripheral-central vision system.

II. Experimental Setup

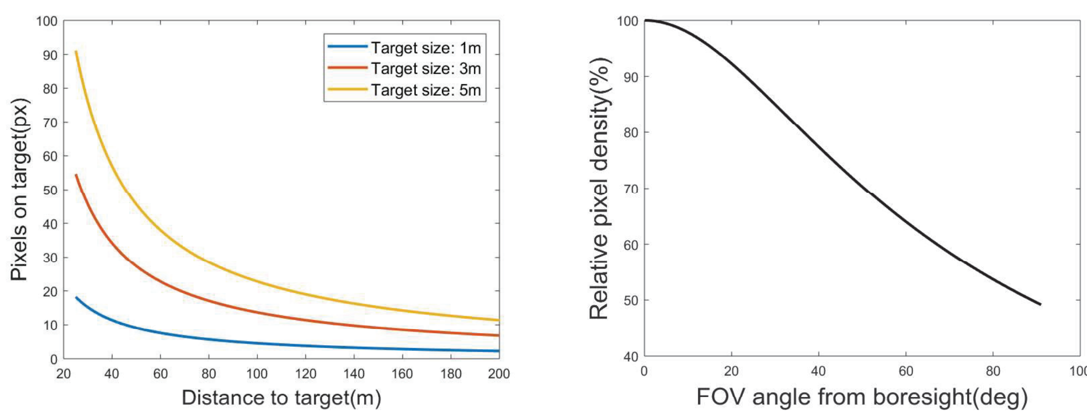
A. Peripheral Vision Camera System

Fast and reliable initial threat detection is crucial for a vision-based ABDAA and counter-UAS systems, so the camera system must be able to see a large area at once. This observation suggests the use of an omnidirectional "peripheral" vision camera for the initial threat detection. We selected the Insta360 Air for a prototype system. The camera has two fisheye lenses, providing 360° coverage in both azimuth and elevation, and is compatible with a variety of embedded hardware systems and with widely used software tools, available through OpenCV and the robot operating system (ROS), for example. Detailed specifications are given in Table 1.

While the peripheral vision camera system provides complete coverage, the images have relatively low resolution and high distortion. The number of pixels associated with an object imaged by the peripheral camera is small because of the large FOV and the limited image resolution. For example, the pixel width of a 5m object at 100m distance is only 22 pixels, as shown in Figure 2(a). Image resolution is less of a concern in threat detection, however, because detection methods such as the optical flow algorithm [9] used here can detect and track even pixel-size moving objects. When there are enough pixels on threat, the peripheral camera can provide additional information, but the lens introduces a high level of image distortion, especially at the edges of the image. Distortion can be partially addressed through proper camera calibration, but the pixel density is unavoidably lower away from the camera boresight. For the InstaAir 360, for example, the pixel density at the image edge is half of that at the image center, as shown in Figure 2(b). In any case, a peripheral vision camera system is quite useful for initial threat detection because of its broad FOV; having detected a threat, a central vision camera may be cued to investigate further. Any additional information that becomes available, e.g., when there are more pixels on threat in the peripheral image, can be combined with central vision imagery to improve overall awareness.

Table 1 Specification of cameras

Parameter	Insta 360 Air	GigE camera
Focal length (mm)	1.0	4.8 - 57.6
Sensor size (mm)	3.3×3.3	4.8×3.6
Pixel size (μm)	2.19×2.19	3.75×3.75
Resolution (px)	$1{,}504 \times 1{,}504$	$1{,}280 \times 960$
Size (mm)	$\phi 36.6 \times 39.6$	$50 \times 50 \times 103$
Weight (g)	26.5	330



(a) Pixels on target against distances from camera (b) Relative pixel density of the undistorted Insta 360 Air image

Fig. 2 Capabilities of Insta360 Air

B. Central Vision Camera System

While the peripheral vision camera system has a large FOV and low resolution, the pan-tilt-zoom (PTZ) central vision camera has a narrower FOV, but a high resolution. The central vision camera is intended to obtain more detailed imagery of a threat detected by the lower resolution peripheral vision camera in order to support classification, pose estimation, and path prediction. For the protype system, a GigE color zoom camera from The Imaging Source was acquired; specifications are shown in Table 1. The GigE camera is mounted on an HDAir Infinity MR S2 gimbal, which enables the camera to be directed toward a cue provided by the peripheral vision camera. Once the central vision camera acquires an image containing the threat, the central camera begins to visually servo on the threat, adjusting zoom and focus, to enabling ranging, classification, pose estimation and path prediction.

(a) Central (left) and peripheral (right) vision cameras

(b) RTK GPS-equipped Hexacopter UAV

Fig. 3 Setup for initial ground-based experiments

C. Ground-based testing

Developmental testing of the prototype system is ongoing, with initial tests aimed at validating predicted imaging performance, integrating the camera systems (e.g., cueing and slewing), and tuning vision algorithms (e.g., detection and visual servoing). The hardware and software setup is based on the ROS framework. Figure 3 shows the cameras mounted on tripods for initial testing in which a mock threat aircraft streams its position, obtained using RTK GPS, to a ground station which computes the necessary gimbal angles and zoom setting to keep the threat within central camera's FOV. Images and IMU data from the central and peripheral cameras are stored along with the threat aircraft's actual position, which serves as ground truth for the localization strategy. The Swiftnav Piksi Multi RTK GPS hardware provides localization accuracy within 5cm in the horizontal plane and within 10cm vertically.

III. Detect

Various computer vision (CV) algorithms have been proposed to detect a moving threat using visual imagery. Attributes that are unique to a particular scenario can pose special challenges or opportunities for visual detection. For a ground-based DAA system, for example, one may find that flying objects appear with reasonable contrast against a static background (e.g., a clear blue or overcast sky). Moreover, non-antagonistic aircraft, including many small UAS, include lighting to make them more visible. Accordingly, in our initial development, we have focused on the use of optical flow for threat detection.

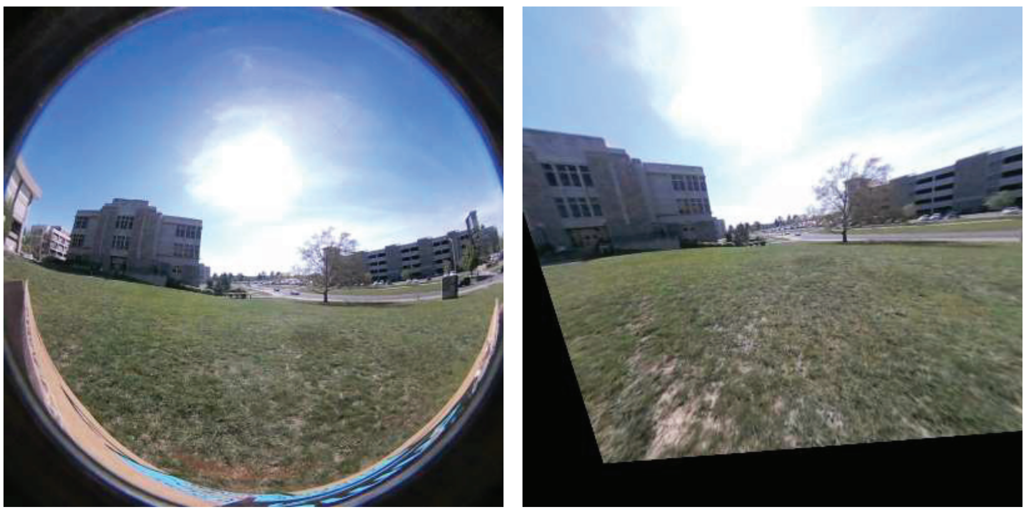
A. Image Stabilization

An optical flow algorithm computes the translational position change of pixels in an image. The pixel coordinates of an object that is moving through an image against a static background change in consecutive images. The pixel velocity is given by the translation vector, which is computed by the optical flow algorithm, enabling the detection of objects moving against a static background. For a camera that is fixed in space, the pixels associated with static objects and with the unmoving background do not move within an image. For an airborne system, however, the camera translates and rotates as the aircraft moves, so that the background and static objects appear to move within the image. Image stabilization is needed to eliminate the apparent motion of static elements in order to detect moving objects.

The homography matrix describes the rotation and translation that relate two images of a given scene. Computing the homography matrix requires the pixel locations of feature points within the two images. After applying image undistortion filters, available in OpenCV [10, 11], to the peripheral camera imagery, the feature points in a given image may be identified in a consecutive image using corner detection [12] and an optical flow algorithm. An estimated homography matrix which has the least number of outliers is then selected using the RANSAC algorithm [13] and is then used to stabilize the next image. Iterating, one obtains a sequence of stabilized images.

Homography-based image stabilization performs quite well in nominal conditions, but the algorithm can be affected by lighting conditions and image noise. As an alternative, one may consider IMU-based image stabilization in which an inertial motion sensor provides information about camera orientation which can then be used for image stabilization similar to the homography-based approach. The approach does require that the image and inertial motion data be accurately synchronized for good performance. The IMU-based approach does not account for the camera's translational motion, however, an effect that is included implicitly in the homography-based approach.

To leverage the strengths of both approaches, a Kalman filter was developed for camera pose data fusion. The fused camera motion data are then converted to a rotation matrix that is used to stabilize images for use in optical flow based detection.



(a) Original peripheral image (b) Undistorted and stabilized peripheral image

Fig. 4 Image undistortion + stabilization

B. Detection Using Optical Flow

In order to detect the threat, we first extract feature points within the image using a corner detector. The optical flow algorithm is then applied to track these feature points in consecutive images. Pixel velocity vectors whose magnitude exceeds a threshold indicate candidate threats. The appropriate threshold differs in different environments and conditions. To explore this sensitivity to conditions, we generated representative "receiver operating characteristic (ROC)" curves for optical flow detection and selected a threshold based on the results. Two ROC curves are shown in Figure 5.

Special challenges arise in vision-based threat detection as proposed, and these are the subject of continuing study. For example, a target coming straight toward the camera might not be detected since the optical flow algorithm works based on only the motion of objects. Also, a cluttered and dynamic background will increase the number of false detections. The problem may be addressed in part by adaptively tuning the detection threshold values and by incorporating complementary feature-based detection algorithms. In this manuscript, we assume the threat is the only moving object in the camera's FOV, but ongoing work is aimed at developing ROC curves for the proposed detection method in varying conditions, including in air-based operations.

After a threat is detected, a Kalman filter is used to provide an estimated look angle to the central vision camera system.

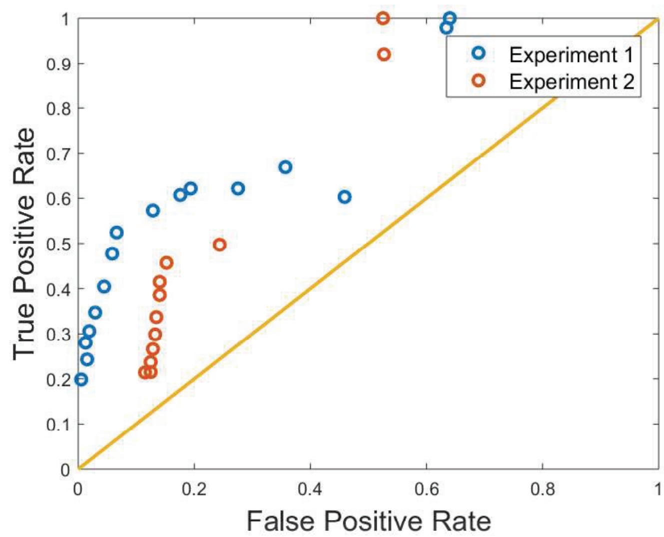


Fig. 5 Receiver operating characteristic(ROC) curve of optical flow detection

When the optical flow algorithm loses track of a threat, the Kalman filter predicts the threat's pixel location based on its last known pixel velocity. If and when a direct measurement of the threat location becomes available once again, the Kalman filter corrects the threat location estimate that serves as a cue to the central vision camera. Figure 6 shows example results for two (undistored) images obtained using the peripheral vision camera with Kalman filtering. Optical flow appears to be effective at detecting candidate threats against a static background, even for threats of small pixel size. For a manageably low number of false detections, one may use the threat location and velocity in pixel coordinates, as described in the next subsections, to cue the central vision camera system that can then classify and estimate the pose of the threat.

Fig. 6 Example results of the optical flow detection with Kalman filter

IV. Stare

A. Obtaining azimuth and elevation to threat

Having detected a threat and obtained its pixel coordinates from the peripheral vision camera system, one may estimate the azimuth and elevation angles to the threat in the peripheral camera-fixed reference frame. These angles are then converted to the central vision camera-fixed reference frame. Given the relative pose between the peripheral ("p") and central ("c") vision cameras, as defined by the proper rotation matrix R^{cp} and the translation vector T^{cp}, the threat vector in the central vision camera-fixed reference frame is

$$\vec{v}^c = \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix} = \begin{bmatrix} R^{cp} \mid T^{cp} \end{bmatrix} \begin{bmatrix} \frac{x^p z^p}{f^p} \\ \frac{y^p z^p}{f^p} \\ z^p \\ 1 \end{bmatrix} \tag{1}$$

where x^p and y^p are the x, y pixel coordinates of the threat on the peripheral image sensor, f^p is the focal length of the peripheral camera, and z^p is the "virtual range" to the threat. The actual range is unknown, but using a virtual range does not cause significant error in the viewing angle estimate when the threat is far away. Here, we analyze the error in the viewing angle cue, for varying values of z^p, in order to determine the most effective value of z^p. The azimuth and elevation angle of threat in the central vision camera-fixed frame are, respectively,

$$\Phi^c = \tan^{-1}\left(\frac{x^c}{z^c}\right) \quad and \quad \Theta^c = \tan^{-1}\left(\frac{y^c}{\sqrt{x^{c2} + z^{c2}}}\right) \tag{2}$$

For the purpose of cueing the central vision camera, we assume the peripheral vision camera initially detects the threat at a sufficiently large distance that one may consider the peripheral and central vision cameras to be collocated. To find a minimum sufficient value of z^p, we consider the error in Φ^c and Θ^p as functions of z^p:

$$\delta\Phi^c = \frac{\partial\Phi^c}{\partial z^p}\delta z^p \quad and \quad \delta\Theta^c = \frac{\partial\Theta^c}{\partial z^p}\delta z^p \tag{3}$$

where δz^p is the error between the actual range z and the virtual range z^p. Figure 7 shows how the error in the computed

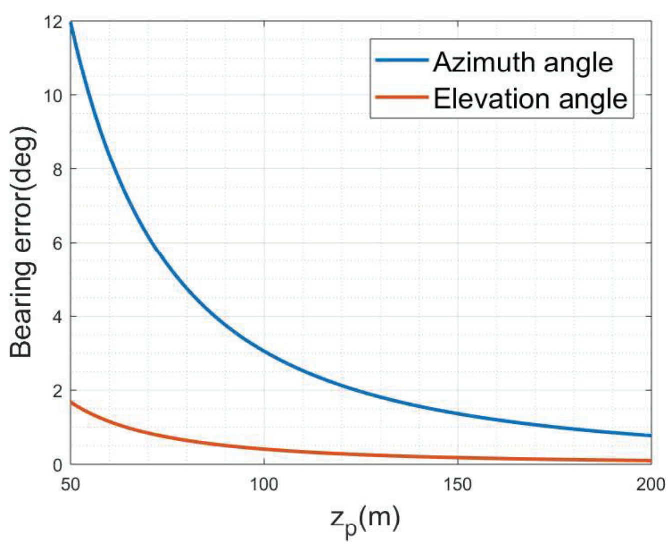


Fig. 7 Variation in angle error with z^p when $\delta z^p = 200$m

threat azimuth and elevation angles vary with z^p, considering a "worst case" deviation $\delta z^p = 200$m. The angle errors $\delta\Phi^c$ and $\delta\Theta^c$ are less than 4° for $z^p > 90$m. Considering the FOV of a typical perspective camera, a 4° pointing error would not remove the threat from the corresponding image, provided the zoom is set sufficiently low. With the selected hardware, our central vision camera has a maximum FOV of about 53°. We therefore take $z^p = 100$m in computing the slew angles for the central vision camera system.

Table 2 Example results of the converted bearing

z (m)	Φ^p(°)	Θ^p(°)	$z^p = 100$**m**		$z^p = z$	
			Φ^c(°)	Θ^c(°)	Φ^c(°)	Θ^c(°)
41.24	25.51	13.63	20.31	12.26	22.24	12.04
55.88	-6.62	11.96	-11.53	11.20	-10.28	11.21
76.28	17.83	9.13	12.74	7.88	13.21	7.85
88.48	17.26	8.21	12.16	6.97	12.36	6.96
102.21	0.86	7.96	-4.10	7.02	-4.13	7.02

Table 2 shows five example results for viewing angles to a given point of interest, as expressed in the peripheral vision camera's reference frame (Φ^p and Θ^p) and as computed for the central vision camera's reference frame using a virtual range $z^p = 100$m and using the true range $z^p = z$. Note that the error between the true and estimated values for Φ^c and Θ^c is smaller than 2° in every case, indicating that the algorithm can cue the viewing angle to the threat with only marginal error.

B. Gimbal Control for Pan and Tilt

The calculated reference azimuth and elevation angles are sent to the gimbal, which uses a PID controller to rotate the camera to the desired attitude. The gimbal controller incorporates data from an IMU attached to the camera and from gimbal axis encoders to provide Kalman filtered orientation feedback. Because the camera's IMU lacks a magnetometer, the gimbal's yaw axis encoder angle is used as the primary azimuth orientation reference during ground testing. The readings from the gimbal's IMU are recorded synchronously, in order to obtain actual camera angles for use in the homography and stabilization algorithms. The quality of the IMU filtered measurements should be accounted for in determining the ranging accuracy. The gimbal manufacturer claims an angular precision of 0.02° in all axes, which has implications for the use of the central camera's zoom capability; a narrower FOV (e.g., fully zoomed) necessitates less error in gimbal orientation in order to maintain view of a threat during tracking. For the current prototype, the minimum horizontal FOV of the central camera is about 4.8°. Because the gimbal precision is significantly finer than this, tracking at full zoom should not present a problem. In addition, upon initial cueing, the central vision camera would start zoomed out to maximize the likelihood of having the threat in its FOV, and zoom in based of strategies discussed in the following subsection.

An important objective is to autonomously cue the gimbal so that the central camera obtains an image of the threat aircraft and then begins to visually track the threat. In preliminary experiments meant to validate range measurements, the gimbal was controlled using position data transmitted directly from the "threat." The locations of the two tripods, supporting the peripheral and central cameras, respectively were surveyed using an RTK GPS unit to establish their offset relative to the RTK base station in an east-north-up (ENU) frame. The threat aircraft's position was then used to calculate and implement the required azimuth and elevation angles.

C. Zoom Control

Several trials using the central and peripheral vision camera system were carried out, some with certain variables held fixed, such as zoom setting or gimbal orientation, to gather a diverse dataset. For the trials making use of the central camera's zoom capability, a control strategy for homing in on the threat was developed. The preliminary implementation in which distance to the threat is obtained directly allows us to map the central camera's zoom setting to the aircraft range based on the desired pixel coverage/frame margin. For example, we can choose to zoom in just enough that the aircraft occupies a certain fraction of the image, allowing a safety buffer to prevent disturbances from causing the threat to escape the image.

Based on desired coverage, one may construct a function that maps a commanded focal length to an actual focal length that the camera is capable of attaining. The central vision camera can change focal length from 4.8mm to 57.6mm, in 100 discrete steps. Since a change in focal length results in varying distortion as the lens shifts, a camera calibration for each focal length setting would be needed to ensure accurate localization. To reduce the required number of calibrations, focal length settings were limited to 10 possible values, spanning the full range of focal lengths.

V. Range

A. Heterogeneous Stereo Vision

Stereo vision algorithms typically assume two, identical perspective cameras. Given the camera baseline – i.e., the perpenduicular distance between two camera boresight axes – the focal length, and the sensor size of the camera, the range to a point can be obtained using triangulation. The range z from the baseline to a feature point using two identical cameras is

$$z = \frac{fb}{X_L - X_R} \tag{4}$$

where X_L and X_R are threat distances on the image sensors and b and f are the baseline and the focal length, respectively.

The equation given is not adequate for a camera with a large FOV, since it has a high level of distortion, and for non-parallel camera boresight axes. Scaramuzza et al. [14] published a camera calibration software for an omnidirectional camera and the software enables conversion from 2D image points on a large FOV camera image to 3D points on the lens. A 3D point that denotes a perceived threat thus defines a vector to that threat in a camera-fixed reference frame. Therefore, a threat position in the global frame can be computed by estimating an intersection of the threat"rays" emanating from two cameras.

If the threat vectors from two cameras ($\vec{v^p}$, $\vec{v^c}$) point a same threat, the intersection of the corresponding rays should exist. However, because of sources of error in computing these threat vectors such as calibration error and pixel error, the two lines might not meet at a point. Therefore, for threat position estimation, the point midway between the two points of closest approach of these two rays are computed using midpoint method [15] for the threat position (Figure 8). The equations for points on each extended line (T^p, T^c) are

$$\begin{aligned} T^p &= O^p + C^p \vec{v^p} \\ T^c &= O^c + C^c \vec{v^c} \end{aligned} \tag{5}$$

where O^p and O^c represent optical center points of the peripheral and central vision cameras, and C^p and C^c are line constants. The closest point is a point which minimizes the distance between T^p and T^c; therefore, C^p and C^c can be estimated by finding constants which minimize $||T^p - T^c||^2$.

B. Error analysis

Every image processing step introduces error in the range estimate. It should be noted that error in the gimbal IMU data of central vision system and pixel error, which results in error of the threat vectors, introduces additional range estimation error, especially for more distant threats. Here, we consider the influence of gimbal IMU error and detection error on the range estimation error.

For the range estimation process, the threat vector in the central vision camera frame is converted to a vector in the peripheral vision camera frame using the gimbal IMU of the central vision system. Thus, the error in the gimbal IMU affects the pixel location of the threat in the image, and this pixel error eventually affects the range estimation error. The pixel error (δx^c,δy^c) due to the gimbal IMU error is

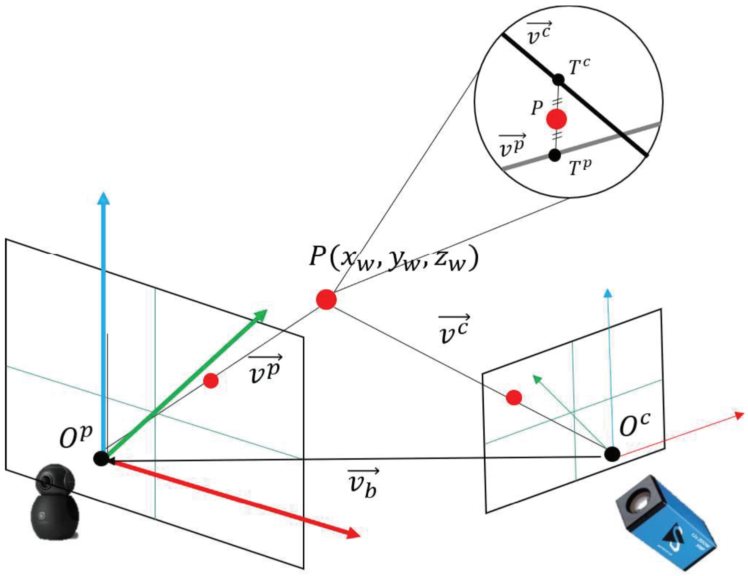


Fig. 8 Threat position estimation using mid-point method

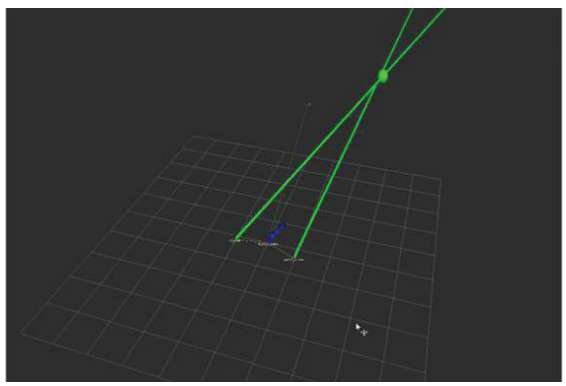


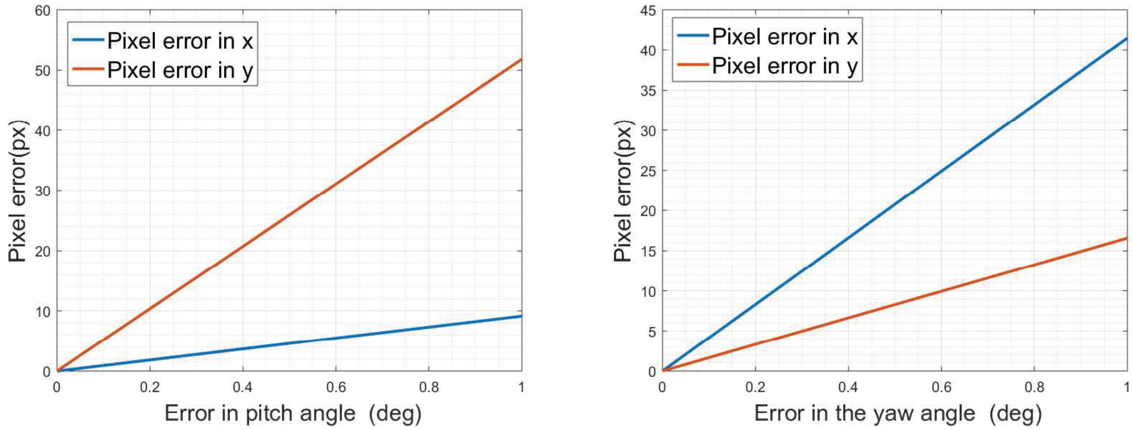
Fig. 9 ROS-based range estimation software

$$
\begin{aligned}
\delta x^c &= \frac{\partial x^c}{\partial \theta}\delta\theta + \frac{\partial x^c}{\partial \psi}\delta\psi \\
\delta y^c &= \frac{\partial y^c}{\partial \theta}\delta\theta + \frac{\partial y^c}{\partial \psi}\delta\psi
\end{aligned}
\tag{6}
$$

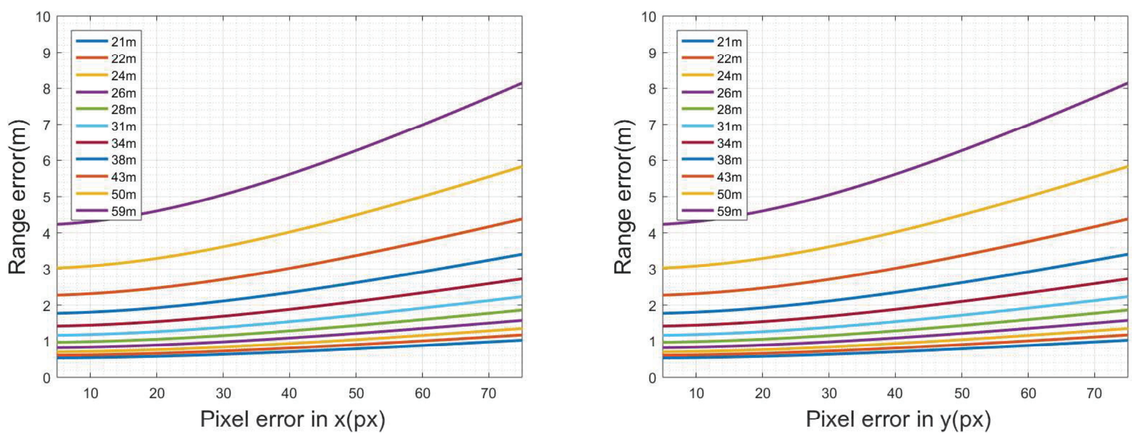
We consider only error in pitch angle (θ) and yaw angle (ψ) of the gimbal IMU since the gimbal we use does not roll. Figure 10(a) depicts the pixel error along with the error in θ and ψ. This pixel error eventually affects the range error. The range error generated by pixel error is

$$
\delta z = \frac{\partial z}{\partial x^c}\delta x^c + \frac{\partial z}{\partial y^c}\delta y^c
\tag{7}
$$

Figure 10(b) illustrates the range error along with pixel error in x and y with various threat ranges. Note that the range error increases with range to the threat. This increase in error can be mitigated by using a longer baseline, if the setup allows it. The analysis verifies that longer baselines and more accurate camera pose estimates reduce range estimation



(a) Pixel error due to IMU error



(b) Range error due to pixel error

Fig. 10 Range estimation error

error and it provides a procedure for estimating the error when constructing a sensing system or assessing a given system's potential performance.

C. Range estimation results

The proposed system estimates the range to a small UAS that is flying within the FOV of both cameras, as described in Section II, using our ROS-based range estimation software; see Figure 9. In preliminary experiments, these range estimates were compared with known values. Figure 11 shows estimation error versus range for a 2m camera baseline. The range error is less than 15m out to a range of about 60m, and tends to increase with range to the threat. This preliminary results also include error, which is occurred by detection error, hence this results still contain lots of error. However, the results still imply that the suggested peripheral-central vision system is capable of estimating the range of the threat, and the error issue will be resolved with false alarm filtering and detection algorithm development in the future works.

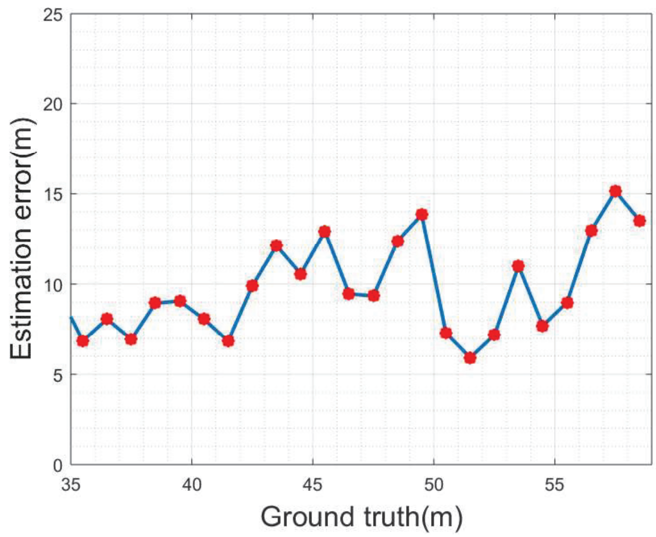


Fig. 11 Range estimation error of experiments

VI. Classify

Object classification can reveal whether a detected object poses a threat and can greatly aid in motion prediction by indicating the dynamic capabilities of a given threat. A threat taxonomy might include coarse categories of threats, such as fixed-wing, helicopter or multirotor, and might also include objects that could be mistaken for threats, such as birds, kites, or floating debris. Finer classifications might include specific models of aircraft.

Several approaches to classification are represented in the literature; the ones explored here employ machine learning and computer vision frameworks. Machine learning, specifically through the use of neural networks (NNs), has been extensively developed for problems involving detection and classification of objects in image frames. Existing deep neural networks such as MobileNet [16] and YOLOv2 [17] have been shown to have high "true positive" rates when trained on sufficiently large and diverse datasets. Both networks can be accelerated to run in real time using GPU resources, however performance depends strongly on the training datasets used.

As a baseline test, YOLOv2 trained on the Common Objects in Context (COCO) data was used to generate bounding boxes around predicted locations of imaged aircraft. Some preliminary test images are shown in Figure 12.

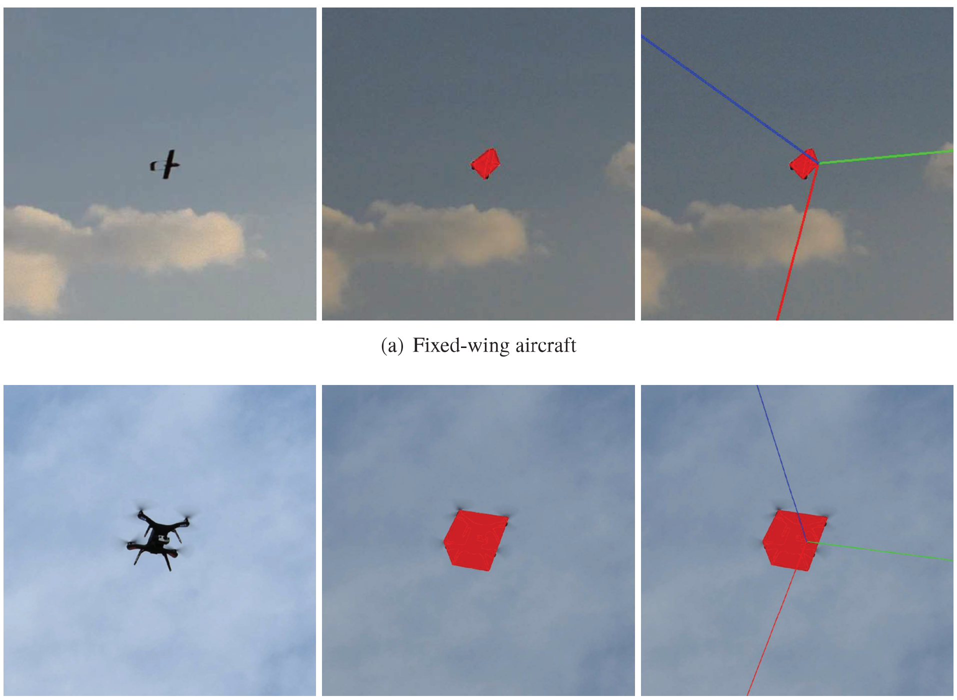
(a) True Positive Sample (b) False Negative Sample (Aircraft present above tree line)

Fig. 12 COCO trained YOLOv2 Samples

Even with a generically trained NN, the aircraft is correctly detected at closer ranges. In edge cases, however, such as when the aircraft descends below the horizon or appears against a less distinct background, the NN tends to fail. Cases with image noise and lower pixel coverage also result in false negatives. A retrained version of YOLO using a dataset containing common classes of small UAS in diverse operating scenarios will presumably produce more robust NN detection and classification. Efforts are under way to generate an initial dataset through experiments and simulations relevant for the flight test scenarios considered in this work; the dataset may be expanded over time, but limited scenarios will be used as a first step towards retraining existing networks.

VII. Predict

Given an image and classification of a threat from the central vision camera, the image is cropped to a region of interest around the threat, and the edge of the threat is extracted using Canny edge detection [18]. Next, the largest polygon containing the identified edges is found (e.g., a pentagon for a fixed-wing aircraft [19, 20]) as shown in Figure 13. The vertices of the polygon are associated with the feature points on the actual model of the threat to compute the pose of the threat aircraft using the POSIT algorithm [21] on the assumption that all the feature points are visible in the image.



(a) Fixed-wing aircraft

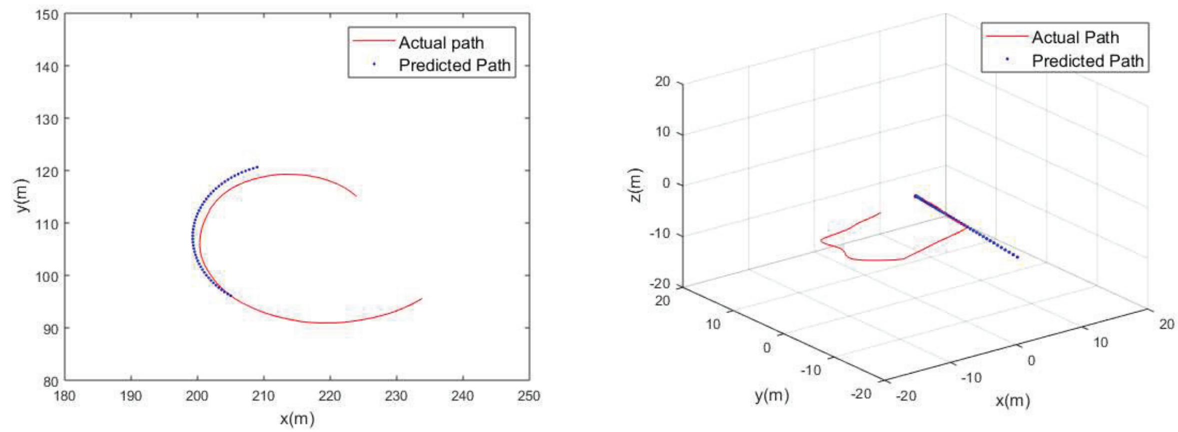
(b) Quadcopter aircraft

Fig. 13 Threat feature point detection and pose estimation

The POSIT algorithm determines a rotation matrix R^{BC} relating a reference frame fixed in the central vision camera to a reference frame fixed in the body of the threat aircraft. The camera orientation R^{CI} relative to the inertial reference frame is known from the camera-fixed IMU. The pose of the threat aircraft may then be determined matrix multiplication:

$$R^{BI} = R^{BC} R^{CI} \tag{8}$$

The motion model of the threat is then used for path prediction. Figure 14 illustrates path prediction simulations for a fixed-wing aircraft and for a quadcopter. The red lines indicate the actual path of the threat aircraft and the blue lines represent the predicted path, based on pose data inferred from visual imagery. In [22], path predictions for a fixed-wing aircraft based on visually inferred pose data were shown to provide quicker and more accurate predictions for a maneuvering aircraft than the predictions based only on threat aircraft position. Ongoing work is aimed at improving path prediction for a larger class of small UAS and on applying the algorithm in real-time.



(a) Fixed-wing aircraft

(b) Quadcopter aircraft

Fig. 14 Path prediction based on the pose

VIII. Conclusion

This paper introduces a peripheral-central vision system for small UAS detection, localization, classification, and prediction. Imagery from the peripheral vision camera are first undistorted and stabilized so that the system may detect moving objects using optic flow and predict their short-motion using a Kalman filter. The viewing angle to the threat is then provided to a central vision camera which slews toward and then visually servos on the threat, automatically zooming in to classify the threat and to estimate its pose. Range is determined using heterogeneous stereo vision. The range estimate can be further improved using the central camera's zoom capability and possibly knowledge of the threat aircraft geometry, as determined through classification. Finally, the path of the threat aircraft can be predicted over a time horizon suitable for DAA or counter-UAS actions by using knowledge of the aircraft class and the real-time pose estimates. Some specific difficulties and limitations arise because the system uses visual imagery and computer vision techniques. These include sensitivity to lighting conditions, background clutter, etc. The advantages of low-cost, light-weight, passive sensing, and the proposed application for short-range, low-altitude scenarios applications might make the system an effective stand-alone solution. Alternatively, the proposed system could enrich the capabilities of complementary active sensors such as RADAR or LiDAR. In particular, the proposed system is capable of providing important data about the threat which other sensors cannot, such as the threat aircraft's class and pose. Ongoing efforts aim to refine and improve the proposed system by enabling real-time threat classification and path prediction from an airborne platform.

References

[1] Drulea, M., Szakats, I., Vatavu, A., and Nedevschi, S., "Omnidirectional stereo vision using fisheye lenses," *In Intelligent Computer Communication and Processing (ICCP), 2014 IEEE International Conference*, 2014, pp. 251–258.

[2] Kita, N., and Kita, Y., "Reference plane based fisheye stereo epipolar rectification," *In 12th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2017, pp. 308–320.

[3] Siewert, S., Andalibi, M., Bruder, S., Gentilini, I., and Buchholz, J., "Drone net architecture for UAS traffic management multi-modal Sensor networking experiments," *In IEEE Aerospace Conference*, 2018.

[4] Laurenzis, M., Hengy, S., Hammer, M., Hommes, A., Johannes, W., Giovanneschi, F., Rassy, O., Bacher, E., Schertzer, S., and Poyet, J. M., "An adaptive sensing approach for the detection of small UAV: first investigation of static sensor network and moving sensor platform," *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVII*, 2018, p. 106460S.

[5] Chen, P., Dang, Y., Liang, R., Zhu, W., and He, X., "Real-time object tracking on a drone with multi-inertial sensing data," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19(1), 2018, pp. 131–139.

[6] Eynard, D., Vasseur, P., Demonceaux, C., and Frémont, V., "UAV altitude estimation by mixed stereoscopic vision," *In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 646–651.

[7] Eynard, D., Demonceaux, C., Vasseur, P., and Fremont, V., "UAV motion estimation using hybrid stereoscopic vision," *In Machine Vision Applications (MVA), 2011 IAPR Conference on*, 2011, pp. 340–343.

[8] Eynard, D., Vasseur, P., Demonceaux, C., and Frémont, V., "Real time UAV altitude, attitude and motion estimation from hybrid stereovision," *Autonomous Robots*, Vol. 33(1-2), 2012, pp. 157–172.

[9] Lucas, B. D., and Kanade, T., "An iterative image registration technique with an application to stereo vision," *Proceedings of the International Joint Conference on Artificial Intelligence*, 1981.

[10] Bradski., G., "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[11] Bradski, G., and Kaehler, A. (eds.), *Learning OpenCV: Computer vision with the OpenCV library*, O'Reilly Media, Inc., 2008.

[12] Harris, C., and Stephens, M., "A combined corner and edge detector," *Alvey vision conference*, Vol. 15, 1988, pp. 10–5244.

[13] Fischler, M. A., and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Vol. 24(6), 1981, pp. 381–395.

[14] Scaramuzza, D., Martinelli, A., and Siegwart, R., "A toolbox for easily calibrating omnidirectional cameras," *In Intelligent Robots and Systems (IROS), 2006 IEEE/RSJ International Conference on*, 2006, pp. 5695–5701.

[15] Beardsley, P., Zisserman, A., and Murray, D., "Navigation using affine structure from motion," *In European Conference on Computer Vision*, 1994, pp. 85–96.

[16] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., "MobileNets: efficient convolutional neural networks for mobile vision applications," *ArXiv e-prints*, 2017.

[17] Redmon, J., and Farhadi, A., "YOLO9000: Better, Faster, Stronger," *arXiv preprint*, 2016.

[18] Canny, J., "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 6, 1986, pp. 679–698.

[19] Persson, A. H., Bondesson, L., and Börlin, N., "Estimation of polygons and areas." *Scandinavian Journal of Statistics*, Vol. 33(3), 2006, pp. 541–559.

[20] Ratschek, H., Rokne, J., and Yap, C., "Exact and optimal convex hulls in 2D," *International Journal of Computational Geometry and Applications*, Vol. 10(2), 2000, p. 109.

[21] Dementhon, D. F., and Davis, L. S., "Model-based object pose in 25 lines of codec," *International Journal of Computer Vision*, Vol. 15(1-2), 1995, pp. 123–141.

[22] Kang, C., Davis, J., Woolsey, C. A., and Choi, S., "Sense and avoid based on visual pose estimation for small UAS," *In Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference*, 2017, pp. 3473–3478.